Chapter 8

Clustering Algorithms

8.1 The *k*-means algorithm

— The *k*-means clustering algorithm is used to find a set of *k* processing unit centers, which represent a local minimum of the total squared Euclidean distances *E* between the *L* exemplars (training patterns) x_i and the nearest of the k centers c_j.

$$E = \sum_{j=1}^{k} \sum_{i=1}^{L} \left\| c_j - x_i \right\|^2$$

<u>The k-means algorithm</u>

Step 1. Initialize:

Choose the number of cluster, k. For each of these k clusters choose an initial cluster center: $\{c_1(m), c_2(m), \ldots, c_k(m)\}$

where $c_j(m)$ represents the value of the cluster center at the *m*th iteration.

Step 2. Distribute samples:

Distribute all sample patterns.

$$x_p \in \theta_j(m) \qquad \text{if } \left\| \mathrm{c}_j(m) - x_p \right\| < \left\| c_i(m) - x_p \right\| \ \text{ for all } \mathrm{i} = 1,2,\ldots,\mathrm{k.\ i} \neq \mathrm{j}$$

where $\theta_j(m)$ represents the population of cluster *j* at iteration *m*.

Step 3. Calculate new cluster center:

$$c_j(m+1) = \frac{1}{M_j} \sum_{x_p \in \theta_j(m)} x_p$$

where M_j is the number of sample patterns attached to θ_j during step 2.

Step 4. Check for convergence:

The condition for convergence is that no cluster has changed its position during step 3.

Example $k = 2$

Pattern [1]=(0, 0)
Pattern [2]=(1, 0)
Pattern [3]=(0, 1)
Pattern [4]=(2, 1)
Pattern [5]=(1, 2)
Pattern [6]=(2, 2)
Pattern [7]=(2, 0)
Pattern [8]=(0, 2)
Pattern [9]=(7, 6)
Pattern [10]=(7, 7)
Pattern [11]=(7, 8)
Pattern [12]=(8, 6)
Pattern [13]=(8, 7)
Pattern [14]=(8, 8)
Pattern [15]=(8, 9)
Pattern [16]=(9, 7)
Pattern [17]=(9, 8)
Pattern [18]=(9, 9)

Step 1. Initialize:

Cluster center [1]=(0, 0)

Cluster center [2]=(1, 0)

Step 2. Distance:

Patterns	Cluster center [1]	Cluster center [2]	Assigned to cluster
Pattern [1]	0.0	1.0	1
Pattern [2]	1.0	0.0	2
Pattern [3]	1.0	1.4	1
Pattern [4]	2.2	1.4	2
Pattern [5]	2.2	2.0	2
Pattern [6]	2.8	2.2	2
Pattern [7]	2.0	1.0	2
Pattern [8]	2.0	2.2	1
Pattern [9]	9.2	8.5	2
Pattern [10]	9.9	9.2	2
Pattern [11]	10.6	10.0	2
Pattern [12]	10.0	9.2	2
Pattern [13]	10.6	9.9	2
Pattern [14]	11.3	10.6	2

Pattern [15]	12.0	11.4	2
Pattern [16]	11.4	10.6	2
Pattern [17]	12.0	11.3	2
Pattern [18]	12.7	12.0	2

Step3. Recalculate the cluster center:

Cluster center [1]= $(\frac{0+0+0}{3}, \frac{0+1+2}{3}) = (0, 1)$

Cluster center [2]=$(\frac{1+2+1+2+...+9}{15}, \frac{0+1+2+2+...+9}{15}) = (5.87, 5.33)$

Iteration 2

Step 2. Distance:

Patterns	Cluster center [1]	Cluster center [2]	Assigned to cluster
Pattern [1]	1.0	7.9	1
Pattern [2]	1.4	7.2	1
Pattern [3]	0.0	7.3	1
Pattern [4]	2.0	5.8	1
Pattern [5]	1.4	5.9	1
Pattern [6]	2.2	5.1	1
Pattern [7]	2.2	6.6	1
Pattern [8]	1.0	6.7	1
Pattern [9]	8.6	1.3	2
Pattern [10]	9.2	2.0	2
Pattern [11]	9.9	2.9	2
Pattern [12]	9.4	2.2	2
Pattern [13]	10.0	2.7	2
Pattern [14]	10.6	3.4	2
Pattern [15]	11.3	4.2	2
Pattern [16]	10.8	3.5	2
Pattern [17]	11.4	4.1	2
Pattern [18]	12.0	4.8	2

Step 3. Recalculate the cluster center:

Cluster center [1] = (1.0, 1.0)

Cluster center [2] = (8.0, 7.5)

Iteration 3

⋮

⋮

Step 3. Recalculate the cluster center:

Cluster center [1] = (1.0, 1.0)

Cluster center [2] = (8.0, 7.5)

Step 4. Check for convergence:

Since the cluster center did not change, the algorithm has converged. Thus the final cluster centers are as follows:

Cluster center [1] = (1.0, 1.0)

Cluster center [2] = (8.0, 7.5)

8.2 The Fuzzy C-means algorithm

—Fuzzy C-means algorithm was powerful data cluster algorithm was introduced by Bezdek (1981).

Fuzzy Clustering

— Consider a finite of the p-dimensional Euclidean space R^p, that is, $x_j \in R^p$ for $j = 1,2,...n.$, the problem is to perform a partition of this collection of patterns into c fuzzy sets with respect to a given criterion, where c is a given number of clusters.

— The criterion is usually to optimizes an *objective function*, which acts as a performance index of clustering.

— The end result of fuzzy clustering can be express by a partition matrix U, such that $U = [u_{ij}]_{i=1,2,..c;\ j=1,2,..,n}$.

where u_{ij} is a numerical value in [0, 1] and expresses the degree to which the element x_j belongs to ith cluster.

— There are two additional constraints on the value of u_{ij}.

(1) A total membership value of the pattern $x_j \in X$ to all classes is equal to 1.

That is, $\sum_{i=1}^{c} u_{ij} = 1$, for all $j = 1,2,..,n$

(2) Every constructed cluster is nonempty and different from the entire set. That is,

$0 < \sum_{j=1}^{n} u_{ij} < n$, for all $j = 1,2,..,n$

— A general form of the fuzzy clustering objective function, denoted J_m as:

$$J_m(u_{ij}, v_k) = \sum_{i=1}^{c}\sum_{j=1}^{k} u_{ij}^m \left\| x_j - v_i \right\|^2, \qquad \text{for } m > 1$$

where v_k is the center vector of the kth cluster, m is called the exponential weight which influences the degree of fuzziness of the membership (partition) matrix.

— A constrained fuzzy partition $\{c_1, c_2, ... c_k\}$ can be a local minimum of the objective function J_m only if the following conditions are satisfied:

➢ $$v_i = \frac{\sum_{j=1}^{n} (u_{ij})^m x_j}{\sum_{j=1}^{n} (u_{ij})^m}, \qquad i = 1,2,...,c. \tag{8.1}$$

➢ $$u_{ij} = \frac{\left(\frac{1}{\left\| x_j - v_k \right\|^2} \right)^{\frac{1}{m-1}}}{\sum_{k=1}^{c} \left(\frac{1}{\left\| x_j - v_k \right\|^2} \right)^{\frac{1}{m-1}}}, \qquad i = 1,2,...,c\, ;\, j = 1,2,...,n. \tag{8.2}$$

Fuzzy C-means Algorithm

Step 1. Fix a number of clusters c $(2 \le c \le n)$ and exponential weight m $(1 \le m \le \infty)$. Choose an initial partition matrix $U^{(0)}$ and termination error ε. Set the iteration index l to zero.

Step 2. Calculate the fuzzy cluster centers $\{ v_i^{(l)} \mid i = 1,2,...,c\}$ by $U^{(l)}$ and formula (8.1).

Step 3. Calculate the new partition matrix $U^{(l+1)}$ by using $\{ v_i^{(l)} \mid i = 1,2,...,c\}$ and formula (8.2).

Step 4. Calculate $\Delta = \left\| U^{(l+1)} - U^{(l)} \right\| = \underset{i,j}{Max} \mid u_{ij}^{(l+1)} - u_{ij}^{(l)} \mid$, if $\Delta > \varepsilon$, then set $l = l + 1$ and got to step 2. If $\Delta \le \varepsilon$, then stop.

Example

A company has three jobs, which the similarity degree of matrix was shown as:

Please clustering these jobs, using the Fuzzy C-means algorithm.

	x_1	x_2	x_3
x_1	1	0.8	0.4
x_2	0.8	1	0.6
x_3	0.4	0.6	1

Step1:

Let $c = 2$, exponential weigh (m)=2, termination error (ε)=0.01, and

the partition matrix, $U^{(0)} = \begin{bmatrix} 0.566 & 0.4 & 0.395 \\ 0.434 & 0.6 & 0.605 \end{bmatrix}$.

Step 2: Calculate the fuzzy cluster centers $\{ v_i^{(l)} \mid i = 1,2,...,c \}$

$$v_i^{(l)} = \frac{\sum_{j=1}^{n} (u_{ij})^m x_j}{\sum_{j=1}^{n} (u_{ij})^m}, \qquad \text{i} = 1,2,..,\text{c}$$

$$v_1^{(0)} = \frac{0.566^2 (1,\ 0.8,\ 0.4) + 0.4^2 (0.8\ ,1,\ 0.6) + 0.395(0.4,\ 0.6,\ 1)}{0.566^2 + 0.4^2 + 0.395^2} = (0.803,\ 0.801,\ 0.597)$$

$$v_2^{(0)} = \frac{0.434^2 (1,\ 0.8,\ 0.4) + 0.6^2 (0.8,\ 1,\ 0.6) + 0.605^2 (0.4,\ 0.6\ ,1)}{0.434^2 + 0.6^2 + 0.605^2} = (0.681,\ 0.799,\ 0.719)$$

Step 3: Calculate the new partition matrix $U^{(l+1)}$

$$u_{ij} = \frac{\left(\frac{1}{\left\| x_j - v_i \right\|^2} \right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c} \left(\frac{1}{\left\| x_j - v_i \right\|^2} \right)^{\frac{1}{m-1}}}, \quad i = 1,2,...,c;\ j = 1,2,...,n.$$

Calculate the distance from each job (x_j) to each cluster center (v_k):

$$\left\|x_j - v_k\right\|^2$$

$$\left\|x_1 - v_1\right\|^2 = (1-0.803)^2 + 0.8-0.801)^2 + (0.4-0.597)^2 = 0.078$$

$$\left\|x_2 - v_1\right\|^2 = (0.8-0.803)^2 + (1-0.801)^2 + (0.6-0.597)^2 = 0.040$$

$$\left\|x_3 - v_1\right\|^2 = (0.4-0.803)^2 + (0.6-0.801)^2 + 1-0.597)^2 = 0.365$$

$$\left\|x_1 - v_2\right\|^2 = (1-0.618)^2 + (0.8-0.799)^2 + (0.4-0.719)^2 = 0.203$$

$$\left\|x_2 - v_2\right\|^2 = (0.8-0.681)^2 + (1-0.799)^2 + (0.6-0.719)^2 = 0.069$$

$$\left\|x_3 - v_2\right\|^2 = (0.4-0.681)^2 + (0.6-0.799)^2 + (1-0.719)^2 = 0.197$$

Calculate the new partition:

$$u_{11}^{(1)} = \frac{\left(\frac{1}{0.078}\right)^{\frac{1}{2-1}}}{\left(\frac{1}{0.078}\right)^{\frac{1}{2-1}} + \left(\frac{1}{0.203}\right)^{\frac{1}{2-1}}} = 0.722$$

$$u_{21}^{(1)} = \frac{\left(\frac{1}{0.203}\right)^{\frac{1}{2-1}}}{\left(\frac{1}{0.078}\right)^{\frac{1}{2-1}} + \left(\frac{1}{0.203}\right)^{\frac{1}{2-1}}} = 0.178$$

$$u_{12}^{(1)} = \frac{\left(\frac{1}{0.040}\right)^{\frac{1}{2-1}}}{\left(\frac{1}{0.040}\right)^{\frac{1}{2-1}} + \left(\frac{1}{0.069}\right)^{\frac{1}{2-1}}} = 0.633$$

$$u_{22}^{(1)} = \frac{\left(\frac{1}{0.069}\right)^{\frac{1}{2-1}}}{\left(\frac{1}{0.040}\right)^{\frac{1}{2-1}} + \left(\frac{1}{0.069}\right)^{\frac{1}{2-1}}} = 0.367$$

$$u_{13}^{(1)} = \frac{\left(\frac{1}{0.365}\right)^{\frac{1}{2-1}}}{\left(\frac{1}{0.365}\right)^{\frac{1}{2-1}} + \left(\frac{1}{0.197}\right)^{\frac{1}{2-1}}} = 0.351$$

$$u_{23}^{(1)} = \frac{\left(\frac{1}{0.197}\right)^{\frac{1}{2-1}}}{\left(\frac{1}{0.365}\right)^{\frac{1}{2-1}} + \left(\frac{1}{0.197}\right)^{\frac{1}{2-1}}} = 0.649$$

Thus, the new partition matrix is:

$$U^1 = \begin{bmatrix} 0.722 & 0.633 & 0.351 \\ 0.278 & 0.367 & 0.649 \end{bmatrix}$$

Step 4: Calculate $\Delta = \left\| U^{(l+1)} - U^{(l)} \right\| = \underset{i,j}{Max} \, | u_{ij}^{(l+1)} - u_{ij}^{(l)} |$

$$U^1 = \begin{bmatrix} 0.722 & 0.633 & 0.351 \\ 0.278 & 0.367 & 0.649 \end{bmatrix} \qquad U^{(0)} = \begin{bmatrix} 0.566 & 0.4 & 0.395 \\ 0.434 & 0.6 & 0.605 \end{bmatrix}$$

$$\begin{aligned} \Delta &= \max\{|0.722-0.566|, |0.633-0.4|, |0.351-0.395|, |0.278-0.434|, |0.367-0.6|, |0.649-0.605|\} \\ &= \max\{0.156, 0.233, 0.044, 0.156, 0.233, 0.044\} \\ &= 0.233 > 0.01 \end{aligned}$$

Go to Step 2

.

.

.

$$U^2 = \begin{bmatrix} 0.896 & 0.881 & 0.116 \\ 0.104 & 0.119 & 0.884 \end{bmatrix}$$

$\Delta = 0.248 > 0.01$

Go to Step 2

.

.

.

$$U^3 = \begin{bmatrix} 0.96 & 0.938 & 0.001 \\ 0.04 & 0.062 & 0.999 \end{bmatrix}$$

$\Delta = 0.115 > 0.01$

Go to Step 2

.

.

.

$$U^4 = \begin{bmatrix} 0.963 & 0.938 & 0.000 \\ 0.037 & 0.062 & 1.000 \end{bmatrix}$$

$\Delta = 0.003 < 0.01$

According to, the result of the partition matrix U^4, the job 1 and job2 belong to cluster1, and the job3 belongs to cluster2.

8.3 SOM

Kohonen Self-Organizing Maps

Kohonen's SOM

-- Unsupervised learning.

-- A topological map.

-- A two-layered network (the two layers are fully interconnected).

-- Usually the second layer is organized as a two-dimensional grid.

-- Random values may be assigned for the initial weights.

-- SOM can be used to cluster a set of continuous-valued vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ into *m* clusters.

Input: $\mathbf{x}$
Weights: $\mathbf{w}_j$
The matching value for each unit j is $\|\mathbf{x} - \mathbf{w}_j\|$.
The unit with the lowest matching value (the best match) wins the competition.

The operation of a Kohonen network

1. Compute a matching value for each unit in the competitive layer.

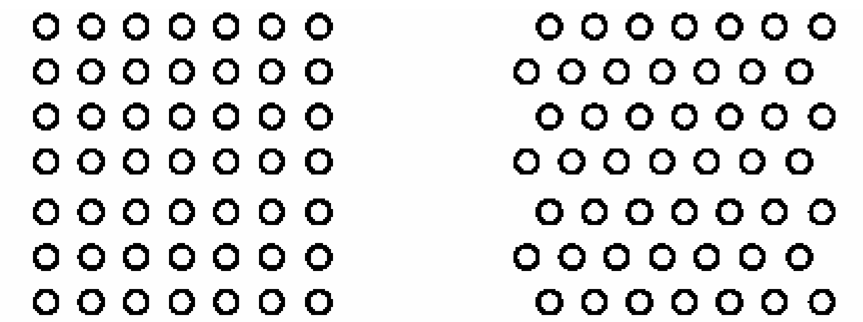
2. Find the best match.

3. Identify the neighborhood around the winner unit.

4. Weights are updated for all units are in the neighborhood of the winning unit.

$$\Delta w_{ij} = \begin{cases} \eta(x_i - w_{ij}) & \text{if unit } j \text{ is in the neighborhood } N_c \\ 0 & \text{otherwise} \end{cases}$$

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}$$



Neighborhoods for rectangular grid   Neighborhoods for hexagonal grid

There are two parameters that must be specified:

1. The value of the learning rate η.

2. The size of the neighborhood N_c.

Example:

$$\eta_0 = 0.2 \sim 0.5$$

$$\eta_t = \eta_0 (1 - \frac{t}{T})$$

$$R_t = R_0 (1 - \frac{t}{T})$$

t: the current training iteration

T: the total number of training iterations to be done.

Typical values for R_0 may be chosen at a half or a third of the width of the competitive layer of processing units.

Note: The learning rate η is a slowly decreasing function of time (or training epochs).

The radius of the neighborhood around a cluster unit also decrease as the clustering process progresses.

Algorithm (Fausett, L, 1994)

Step 1: Initialize weights w_{ij}

Set topological neighborhood and learning rate parameters.

Step 2: While stopping condition is false, do Steps 3-9.

Step 3: For each input vector **x**, do Steps 4-6.

Step 4: For each j, compute:

$$D(j) = \sum_i (w_{ij} - x_i)^2$$

Step 5: Find index j such that $D(j)$ is a minimum.

Step 6: For all unit j within a specified neighborhood of j, and for all i:

$$w_{ij}^{new} = w_{ij}^{old} + \eta(x_i - w_{ij}^{old})$$

Step 7: Update learning rate.

Step 8: Reduce radius of topological neighborhood at specified times.

Step 9: Test stopping condition.

Simple example

For example, A SOM to cluster four vectors

Let the vector to be clustered be

(1,1,0,0); (0,0,0,1); (1,0,0,0) and (0,0,1,1).

The maximum number of cluster to be formed is $m = 2$.

Suppose the learning is $\eta(0) = 0.6, \eta(\mathrm{t}+1) = 0.5\eta(\mathrm{t})$.

Only the winning unit is allowed to learn, i.e., $R = 0$.

Step 1: initial weight matrix:

$$\begin{bmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \\ 0.5 & 0.7 \\ 0.9 & 0.3 \end{bmatrix}$$

Initial radius: $R = 0$. Initial learning rate: $\eta(0) = 0.6$.

For the first vector (1,1,0,0),

$$D(1) = (0.2-1)^2 + (0.6-1)^2 + (0.5-0)^2 + (0.9-0)^2 = 1.86;$$
$$D(2) = (0.8-1)^2 + (0.4-1)^2 + (0.7-0)^2 + (0.3-0)^2 = 0.98.$$

The input vector is closest to output node 2, so $j = 2$.

The weights on the winning unit are updated:

$$w_{i2}^{new} = w_{i2}^{old} + \eta(0)(x_i - w_{i2}^{old})$$

$$= \begin{bmatrix} 0.8 \\ 0.4 \\ 0.7 \\ 0.3 \end{bmatrix} + (0.6)\begin{bmatrix} 0.2 \\ 0.6 \\ -0.7 \\ -0.3 \end{bmatrix} = \begin{bmatrix} 0.92 \\ 0.76 \\ 0.28 \\ 0.12 \end{bmatrix}$$

This gives the weight matrix:

$$\begin{bmatrix} 0.2 & 0.92 \\ 0.6 & 0.76 \\ 0.5 & 0.28 \\ 0.9 & 0.12 \end{bmatrix}$$

$$\vdots$$
$$\vdots$$
$$\vdots$$

$$\begin{bmatrix} 0.0 & 1.0 \\ 0.0 & 0.5 \\ 0.5 & 0.0 \\ 1.0 & 0.0 \end{bmatrix}$$